136 by controlling the persistent thread/process 142. Claim 11, and its associated dependent claims 12 and 13, simply involve indirect initiation of the test script via the control program as in the FIG. 4B example. Applicants therefore submit that there is no inconsistency between dependent claim 11 and independent claim 1. The §112 rejection is believed to be improper and should be withdrawn.

Independent claims 1, 17 and 18 stand rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,901,315 to Edwards et al. (hereinafter "Edwards").

Applicants initially note that MPEP §2131 specifies that a given claim is anticipated "only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference," citing Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, MPEP §2131 indicates that the cited reference must show the "identical invention . . . in as complete detail as is contained in the . . . claim," citing Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). For the reasons identified below, Applicants submit that the Examiner has failed to establish anticipation of claims 1, 17 and 18 by Edwards.

Independent claim 1, by way of example, is directed to a method for use in providing improved fault tolerance in a computing system comprising at least one computing machine. The method includes the steps of executing a control program in conjunction with a fault tolerance software system running on the at least one computing machine, and initiating via the control program a test script program which sends one or more requests to a monitored program, wherein the test script program processes corresponding responses to the one or more requests, and generates at least one return value utilizable by the control program to indicate a failure condition in the monitored program.

Applicants submit that the Edwards reference not only fails to meet the above-described limitations of claim 1, but is actually non-analogous art relative to the present invention. As indicated above, the present invention relates to the field of fault tolerant computing systems, which generally involves deployed computing systems that are configured so as to be able to continue to operate in the presence of program failures, machine failures or other types of system faults. See the

specification at, for example, page 1, lines 10-27. The Edwards reference is an entirely different field, namely, the field of software debugging tools, which are tools used in a software design phase to facilitate the locating of problems or other "bugs" in a piece of software code that is being designed. For example, Edwards describes the operation of debugger 19, which the Examiner alleges is anticipatory of claim 1, as follows:

> The debugger 19 includes a graphical user interface (GUI) or "front end" 21, a debug engine or "back end" 22, and a probe 24. The GUI of the front end (FE) provides a user-friendly interface through which the user makes debug requests and views the status of the "target" application being debugged. The GUI may be implemented through the existing Windows-based display interface of the computer on which the debugger runs. Generally, the debug engine (DE) 22 constituting the back end (BE) performs all of the debugging work. As will be seen in more detail below, the probe 24 is used by the debug engine (DE) to abstract communications (since the target application being debugged may run on the same or a different machine), to manipulate the target application, and to report so-called debug "events" to the DE. As is well-known, representative debug "events" comprise breakpoint events, single step events, a load events (which may include dll events) and/or fault events. The function of a debugger is to identify such event(s) in the target application (i.e. the application being debugged) and report those events back to the user.

This describes nothing more than a conventional software debugging tool, and is entirely unrelated to the field of fault tolerant computing. To put it another way, the present invention relates to techniques for configuring a computing system such that the system can continue to operate in the presence of faults, while Edwards relates to a tool for identifying and removing bugs from a piece of software code that is being designed. Thus, Edwards fails to teach or suggest a fault tolerance software system as recited in claim 1, and similarly fails to teach or suggest a control program that executes in conjunction with a fault tolerance software system.

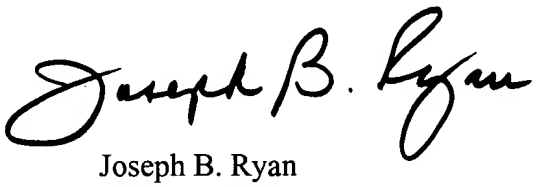
Since Edwards fails to meet at least one limitation of independent claim 1, and also fails to provide its associated advantages in terms of improved fault tolerant computing, that claim is not anticipated by Edwards.

Independent claims 17 and 18 include limitations similar to those of claim 1, and are believed allowable for substantially the same reasons that claim 1 is believed allowable.

Dependent claims 2-16 are believed allowable for at least the reasons given above with regard to independent claim 1, and are also believed to define separately-patentable subject matter over Edwards.

In view of the above, Applicants believe that claims 1-18 are in condition for allowance, and respectfully request withdrawal of the §112, §102(b) and §103(a) rejections.

Respectfully submitted,



Date: August 5, 2004

Joseph B. Ryan
Attorney for Applicant(s)
Reg. No. 37,922
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-7517